Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	3	(clean adj invalidat$4) same (re-validat$3 revalidat$3) same ((dirty invalidat$3) with (line entry data))	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/02/16 11:09
L3	3	((cach$3 near4 (line entry data)) with (pre-load$3 pre-own$4 pre-read$3)) and ((invalidat$3 with (table memory buffer register queu$4 directory list storage)) same ownership)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/02/16 11:15
L4	46	(re-validat$3 revalidat$3) same ((dirty invalidat$3) with (line entry data))	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/02/16 11:10
L5	11645	((invalidat$3 in-validat$3 dirty) with (buffer register history table queu$4 memory storage directory list)) and4	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/02/16 11:11
L6	43	((invalidat$3 in-validat$3 dirty) with (buffer register history table queu$4 memory storage directory list)) and 4	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/02/16 11:11
L7	43	5 and 6	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/02/16 11:16
L8	969	711/141.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/02/16 11:15
L9	4	7 and 8	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2005/02/16 11:16

PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+(invalidate) +and +((entry) +or +(line) +or +(data)) +and

SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before May 2000
Terms used
invalidate and **entry** or **line** or **data** and **history** **table** and **cache** **revalidate** **revalidation**

Found **86** of **104,754**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 86

Result page: **1** 2 3 4 5 next

Relevance scale

1 A high-level abstraction of shared accesses
Peter J. Keleher
February 2000 **ACM Transactions on Computer Systems (TOCS)**, Volume 18 Issue 1

Full text available: pdf(183.57 KB) Additional Information: full citation, abstract, references, index terms, review

We describe the design and use of the tape mechanism, a new high-level abstraction of accesses to shared data for software DSMs. Tapes consolidate and generalize a number of recent protocol optimizations, including update-based locks and recorded-replay barriers. Tapes are usually created by "recording" shared accesses. The resulting recordings can be used to anticipate future accesses by tailoring data movement to application semantics. Tapes-based mechanisms a ...

Keywords: DSM, programming libraries, shared memory, update protocols

2 Scale and performance in a distributed file system
John H. Howard, Michael L. Kazar, Sherri G. Menees, David A. Nichols, M. Satyanarayanan, Robert N. Sidebotham, Michael J. West
February 1988 **ACM Transactions on Computer Systems (TOCS)**, Volume 6 Issue 1

Full text available: pdf(2.38 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The Andrew File System is a location-transparent distributed tile system that will eventually span more than 5000 workstations at Carnegie Mellon University. Large scale affects performance and complicates system operation. In this paper we present observations of a prototype implementation, motivate changes in the areas of cache validation, server process structure, name translation, and low-level storage representation, and quantitatively demonstrate Andrews ability to scale gracefully. W ...

3 Cache Memories
Alan Jay Smith
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3

Full text available: pdf(4.61 MB) Additional Information: full citation, references, citings, index terms

4 Experimental comparison of memory management policies for NUMA multiprocessors
Richard P. Larowe, Carla Schlatter Ellis
November 1991 **ACM Transactions on Computer Systems (TOCS)**, Volume 9 Issue 4
Full text available: pdf(3.17 MB) Additional Information: full citation, references, citings, index terms

5 4.2BSD and 4.3BSD as examples of the UNIX system
John S. Quarterman, Abraham Silberschatz, James L. Peterson
December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4
Full text available: pdf(4.07 MB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper presents an in-depth examination of the 4.2 Berkeley Software Distribution, Virtual VAX-11 Version (4.2BSD), which is a version of the UNIX Time-Sharing System. There are notes throughout on 4.3BSD, the forthcoming system from the University of California at Berkeley. We trace the historical development of the UNIX system from its conception in 1969 until today, and describe the design principles that have guided this development. We then present the internal data structures and ...

6 Apologizing versus asking permission: optimistic concurrency control for abstract data types
M. Herlihy
March 1990 **ACM Transactions on Database Systems (TODS)**, Volume 15 Issue 1
Full text available: pdf(2.38 MB) Additional Information: full citation, abstract, references, citings, index terms

An optimistic concurrency control technique is one that allows transactions to execute without synchronization, relying on commit-time validation to ensure serializability. Several new optimistic concurrency control techniques for objects in decentralized distributed systems are described here, their correctness and optimality properties are proved, and the circumstances under which each is likely to be useful are characterized. Unlike many methods that classify operations only a ...

7 Tolerating latency in multiprocessors through compiler-inserted prefetching
Todd C. Mowry
February 1998 **ACM Transactions on Computer Systems (TOCS)**, Volume 16 Issue 1
Full text available: pdf(410.70 KB) Additional Information: full citation, abstract, references, citings, index terms, review

The large latency of memory accesses in large-scale shared-memory multiprocessors is a key obstacle to achieving high processor utilization. Software-controlled prefetching is a technique for tolerating memory latency by explicitly executing instructions to move data close to the processor before the data are actually needed. To minimize the burden on the programmer, compiler support is needed to automatically insert prefetch instructions into the code. A key challenge when ...

Keywords: compiler optimization, prefetching

8 Verification techniques for cache coherence protocols
Fong Pong, Michel Dubois
March 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 1
Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references, citings, index

terms

In this article we present a comprehensive survey of various approaches for the verification of cache coherence protocols based on state enumeration, (symbolic model checking, and symbolic state models. Since these techniques search the state space of the protocol exhaustively, the amount of memory required to manipulate that state information and the verification time grow very fast with the number of processors and the complexity of the protocol mechanism ...

Keywords: cache coherence, finite state machine, protocol verification, shared-memory multiprocessors, state representation and expansion

9 Data replicas in distributed information services
H. M. Gladney
March 1989 **ACM Transactions on Database Systems (TODS)**, Volume 14 Issue 1

Full text available: pdf(1.94 MB) Additional Information: full citation, abstract, references, index terms, review

In an information distribution network in which records are repeatedly read, it is cost-effective to keep read-only copies in work locations. This paper presents a method of updating replicas that need not be immediately synchronized with the source data or with each other. The method allows an arbitrary mapping from source records to replica records. It is fail-safe, maximizes workstation autonomy, and is well suited to a network with slow, unreliable, and/or expensive communications links ...

10 Query evaluation techniques for large databases
Goetz Graefe
June 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 2

Full text available: pdf(9.37 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Database management systems will continue to manage large data volumes. Thus, efficient algorithms for accessing and manipulating large sets and sequences will be required to provide acceptable performance. The advent of object-oriented and extensible database systems will not solve this problem. On the contrary, modern data models exacerbate the problem: In order to manipulate large sets of complex objects as efficiently as today's database systems manipulate simple records, query-processi ...

Keywords: complex query evaluation plans, dynamic query evaluation plans, extensible database systems, iterators, object-oriented database systems, operator model of parallelization, parallel algorithms, relational database systems, set-matching algorithms, sort-hash duality

11 Compiler transformations for high-performance computing
David F. Bacon, Susan L. Graham, Oliver J. Sharp
December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms, review

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis of scalar quantities and data-flow techniques. In contrast, optimizations for high-performance superscalar, vector, and parallel processors maximize parallelism and memory locality with transformations that rely on tracking the properties o ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimization, parallelism, superscalar processors, vectorization

12 A structural view of the Cedar programming environment
Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann
August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

13 Technique for automatically correcting words in text
Karen Kukich
December 1992 **ACM Computing Surveys (CSUR)**, Volume 24 Issue 4

Full text available: pdf(6.23 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Research aimed at correcting words in text has focused on three progressively more difficult problems:(1) nonword error detection; (2) isolated-word error correction; and (3) context-dependent work correction. In response to the first problem, efficient pattern-matching and n-gram analysis techniques have been developed for detecting strings that do not appear in a given word list. In response to the second problem, a variety of general and application-specific spelling cor ...

Keywords: n-gram analysis, Optical Character Recognition (OCR), context-dependent spelling correction, grammar checking, natural-language-processing models, neural net classifiers, spell checking, spelling error detection, spelling error patterns, statistical-language models, word recognition and correction

14 Hints for computer system design
Butler W. Lampson
October 1983 **ACM SIGOPS Operating Systems Review , Proceedings of the ninth ACM symposium on Operating systems principles**, Volume 17 Issue 5

Full text available: pdf(1.73 MB) Additional Information: full citation, abstract, references, citings, index terms

Experience with the design and implementation of a number of computer systems, and study of many other systems, has led to some general hints for system design which are described here. They are illustrated by a number of examples, ranging from hardware such as the Alto and the Dorado to applications programs such as Bravo and Star.

15 Informing memory operations: memory performance feedback mechanisms and their applications
Mark Horowitz, Margaret Martonosi, Todd C. Mowry, Michael D. Smith
May 1998 **ACM Transactions on Computer Systems (TOCS)**, Volume 16 Issue 2

Full text available: pdf(344.74 KB) Additional Information: full citation, abstract, references, index terms, review

Memory latency is an important bottleneck in system performance that cannot be adequately solved by hardware alone. Several promising software techniques have been shown to address this problem successfully in specific situations. However, the generality of these software approaches has been limited because current architecturtes do not provide a fine-grained, low-overhead mechanism for observing and reacting to memory behavior directly. To fill this need, this article proposes a new class ...

Keywords: cache miss notification, memory latency, processor architecture

16 Adaptive data prefetching using cache information
Ando Ki, Alan E. Knowles
July 1997 **Proceedings of the 11th international conference on Supercomputing**
Full text available: pdf(1.89 MB) Additional Information: full citation, references, citings, index terms

17 Hardware-assisted replay of multiprocessor programs
David F. Bacon, Seth Copen Goldstein
December 1991 **ACM SIGPLAN Notices , Proceedings of the 1991 ACM/ONR workshop on Parallel and distributed debugging**, Volume 26 Issue 12
Full text available: pdf(1.20 MB) Additional Information: full citation, references, citings, index terms

18 A structured approach for the definition of the semantics of active databases
Piero Fraternali, Letizia Tanca
December 1995 **ACM Transactions on Database Systems (TODS)**, Volume 20 Issue 4
Full text available: pdf(4.15 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Active DBMSs couple database technology with rule-based programming to achieve the capability of reaction to database (and possibly external) stimuli, called events. The reactive capabilities of active databases are useful for a wide spectrum of applications, including security, view materialization, integrity checking and enforcement, or heterogeneous database integration, which makes this technology very promising for the near future. An active database system consists of ...

Keywords: active database systems, database rule processing, events, fixpoint semantics, rules, semantics

19 Principles of transaction-oriented database recovery
Theo Haerder, Andreas Reuter
December 1983 **ACM Computing Surveys (CSUR)**, Volume 15 Issue 4
Full text available: pdf(2.48 MB) Additional Information: full citation, references, citings, index terms, review

20 Compiling nested data-parallel programs for shared-memory multiprocessors
Siddhartha Chatterjee
July 1993 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 15 Issue 3
Full text available: pdf(4.17 MB) Additional Information: full citation, references, citings, index terms, review

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE *Xplore*®
RELEASE 1.8

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review

Quick Links

» Se

Welcome to IEEE *Xplore*®

- ○- Home
- ○- What Can I Access?
- ○- Log-out

Tables of Contents

- ○- Journals & Magazines
- ○- Conference Proceedings
- ○- Standards

Search

- ○- By Author
- ○- Basic
- ○- Advanced
- ○- CrossRef

Member Services

- ○- Join IEEE
- ○- Establish IEEE Web Account
- ○- Access the IEEE Member Digital Library

IEEE Enterprise

- ○- Access the IEEE Enterprise File Cabinet

Your search matched **0** of **1128145** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri
new one in the text box.

invalidate <and>revalidation<and>history **Search**

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:
No documents matched your query.

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top